AFRL-AFOSR-VA-TR-2016-0071

TYPE THEORY, COMPUTATION AND INTERACTIVE THEOREM PROVING

Jeremy Avigad
CARNEGIE MELLON UNIVERSITY

09/01/2015
Final Report

DISTRIBUTION A: Distribution approved for public release.

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTA2
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to the Department of Defense, Executive Service Directorate (0704-0188). Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
30-08-2015	FINAL	15 JULY 2012 to 14 JULY 2015

4. TITLE AND SUBTITLE FINAL REPORT FOR AWARD: TYPE THEORY, COMPUTATION, AND INTERACTIVE THEOREM PROVING	**5a. CONTRACT NUMBER** (not applicable)
	5b. GRANT NUMBER FA9550-12-1-0370
	5c. PROGRAM ELEMENT NUMBER (not applicable)
6. AUTHOR(S) DR. JEREMY AVIGAD AND DR. ROBERT HARPER	**5d. PROJECT NUMBER** (not applicable)
	5e. TASK NUMBER (not applicable)
	5f. WORK UNIT NUMBER (not applicable)
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) CARNEGIE MELLON UNIVERSITY OFFICE OF SPONSORED PROJECTS 5000 FORBES AVE PITTSBURGH PA 15213-3815 (412) 268-8746	**8. PERFORMING ORGANIZATION REPORT NUMBER** (none)
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) USAF, AFRL AF OFFICE OF SCIENTIFIC RESEARCH 875 N RANDOLPH ST. ROOM 3112 ARLINGTON VA 22203	**10. SPONSOR/MONITOR'S ACRONYM(S)** CMU
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) (not known)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A, approved for public release

13. SUPPLEMENTARY NOTES
(not applicable)

14. ABSTRACT
Type theory is a logical formalism that is rich enough to express complex mathematical and computational assertions. In this project, Avigad and Harper developed type-theoretic algorithms and formalisms that can support the development of secure and reusable software libraries, as well as the development of methods of automated reasoning in mathematics and libraries of mathematical knowledge.

15. SUBJECT TERMS
type theory, interactive theorem proving, program verification, homotopy type theory, automated reasoning

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Jeremy Avigad
U	U	U	UU	6	**19b. TELEPHONE NUMBER** *(Include area code)* 412-268-8149

Reset

DISTRIBUTION A: Distribution approved for public release

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

Type Theory, Computation, and Interactive Theorem Proving

Jeremy Avigad and Robert Harper

August 28, 2015

1 Statement of Objectives

Seminal contributions of mathematical logic in the twentieth century include formal models of computation and proof. We now know that that mathematical proof can, in principle, be reduced to a small number of basic axioms and rules, and that computation can be understood in terms of a small number of primitive operations. But there is a tremendous gap between our high-level understanding of algorithms and proofs and these low-level implementations. This project developed logical methods, centered on formal type theory, to bridge the gap.

Specifically, the project aimed to develop applications of logic and type theory to the formal verification of mathematical knowledge and the development of systems of secure and flexible computation. The two are intimately related: mathematical formalisms are used to verify properties of computational systems, and, conversely, computation is central to mathematics. The project combined Harper's expertise in the formal semantics of computation and Avigad's expertise in interactive theorem proving in a synergistic way. The project was divided into two tracks, corresponding to these two areas of focus.

The first track, designed by Harper, dealt with intuitionistic type theory, which is a comprehensive foundation for programming that integrates language design, program development, and program verification. A type is defined by specifying its elements and the ways in which we may compute with them, and by specifying when two elements are to be considered equal. A programming language is, therefore, a collection of types that comprise its computational capabilities. By the identification of propositions with types, we obtain both a computational interpretation of proofs as programs, and an integration of verification with programming by type checking. Critical to this integration is the concept of a dependent type, which allows for the expression of precise properties of programs and data. Harper pursued powerful new directions based on these ideas.

The second track, designed by Avigad, dealt with interactive theorem proving, an important method of verifying complex mathematical theorems and algorithms. This has further applications in the verification of complex hardware

DISTRIBUTION A: Distribution approved for public release

and software systems, which are typically described in mathematical terms. Avigad contributed to the theoretical and practical understanding needed to provide better automated support and infrastructure for interactive theorem proving systems. Specifically, he contributed to contemporary methods and technology in verifying theorems of real analysis, measure-theoretic probability, algebra, and algebraic topology, and push the boundaries of what can be achieved with these methods. He also worked to develop better decision procedures and search procedures, and obtain better theoretical and logical formulations of the kinds of reasoning that are effective in these domains.

The next two sections describe some of the highlights of the research carried out in each track.

2 Track 1: Type theoretic foundations

Harper participated in the Univalent Foundations Program at IAS for the development and application of homotopy type theory. Harper's student Kuen-bang Hou worked on the mechanization of the Seifer-van Kampen Theorem and the Blakers-Massey theorem in HOTT, and his student Carlo Angiuli worked with him and others at the IAS on the computational interpretation of homotopy type theory. Harper's post-doc Dan Licata co-developed the main results in homotopy theory as formulated in HoTT. Harper also initiated investigation into cohomology in HOTT by Angiuli, Licata and student Evan Cavallo. The Agda library for homotopy type theory was developed chiefly by Angiuli, Cavallo, Hou, and Licata.

Harper, in collaboration with student Carlo Angiuli, postdoctoral fellow Edward Morehouse, and collaborator Daniel Licata, devised a homotopical theory of revision control in source code management systems. This provides an application of homotopy type theory to a natural problem in computer science, and encouraged the development of an Agda library for reasoning about higher-dimensional paths using a type-theoretic concept of a cube (heterogeneous identification) formulated by Licata. This same library was used by student Evan Cavallo, in collaboration with Carlo Angiuli and Harper, in the proof of the Meyer-Vietoris Theorem in homotopy type theory, an important step in the development of cohomology in type theory. Harper and student Kuen-Bang Hou developed a machine-checked proof of the equivalence of group actions and covering spaces in homotopy type theory, again demonstrating the simplicity and elegance of homotopy type theory for expressing and checking proofs of known results in algebraic topology.

Student Joe Tassarotti developed a mechanized proof of the Read-Copy-Update algorithm used in the Linux kernel using a logic for reasoning about weak memory models formulated in Coq. The RCU algorithm is a very intricate algorithm relying on careful pointer manipulation whose correctness relies on an adversarial analysis of all possible concurrency scenarios. Doing so is especially difficult when working with modern computer architectures that provide only very weak guarantees about how concurrently executing processes may view

DISTRIBUTION A: Distribution approved for public release

updates made by another. The result represents a high-water mark in practical verification of concurrent programs using mechanized provers.

3 Track 2: Interactive theorem proving and automated reasoning

3.1 Homotopy type theory

Avigad participated in the Univalent Foundations Program at IAS for the development and application of homotopy type theory. He worked with Krzysztof Kapulkin and Peter Lumsdaine to formalize properties of homotopy limits in the Coq theorem prover. In other words, they developed an extensive theory of categorical limits over diagrams where diagram identities only hold up to homotopy, rather than "on the nose." This resulted in publication [15].

Avigad also advised two graduate students in work on homotopy type theory. Jakob von Raumer, an MS from Technische Universität Karlsruhe, wrote his thesis under Avigad's supervision [17]. The thesis project consists of a formalization of results in non-abelian topology, using the Lean theorem prover, in the framework of homotopy type theory. The results are especially notable for the complexity of the algebraic structures he dealt with. Second, Avigad has served as advisor to Floris van Doorn, who has contributed seminal results to the new interactive theorem prover, Lean, and has begun to develop a formally verified theory of higher-inductive types and reductions between them.

3.2 Polya: a heuristic theorem prover for real-valued inequalities

Avigad worked with Ph.D. student Rob Lewis, and postdoc Cody Roux, to develop new methods of verifying real-valued inequalities automatically. They developed a prototype implementation in Python [8] (an expanded version of the paper will apear in an issue of the *Journal of Automated Reasoning* featuring the best papers from that conference). Avigad and Lewis have been developing extensions to the system, and have begun work on a formal implementation in the Lean theorem prover (see below). The system performed suprisingly well on over 4,000 benchmark problems developed by André Platzer, in connection with his KeyMaera system for verifying hybrid systems, and will be used as a back-end to that system.

In the spring of 2015, Avigad and Lewis visited Jon Borwein (University of Newcastle, and the center for Computer-Assisted Research Mathematics and its Applications). There, Avigad delivered a series of lectures on formal methods in mathematics, and Avigad, Borwein, and Lewis began to discuss ways of incorporating methods of convex analysis to Polya's heuristics.

DISTRIBUTION A: Distribution approved for public release

3.3 The *Lean* theorem prover

Avigad has contributed to the development of Lean, a new open-source interactive theorem prover developed under the leadership of Leonardo de Moura at Microsoft (`http://leanprover.github.io/`, see also [12, 13]). He has led the development of Lean's standard library, and has supervised student work on the project: the development of the HoTT library by Floris van Doorn, the development of the theory of ordered fields and the reals by Rob Lewis, the development of the theory of lists by undergraduate student Parikshit Khanna (IIT Kanpur), and work on coinductive types by undergraduate students Ken Sakayori (University of Tokyo).

Avigad, de Moura, and Kong have developed an online interactive theorem prover and have written an online tutorial *Theorem Proving in Lean* [14], `https://leanprover.github.io/tutorial/tutorial.pdf`. Avigad taught a graduate seminar on the Lean in the spring of 2012 (`http://leanprover.github.io/cmu-15815-s15/`) and will teach an undergraduate introduction to logic using Lean in the fall of 2015. Reaction to the system and the tutorial have been strongly positive.

3.4 Formally verified mathematics

In addition to developing the Lean library, Avigad completed a formalization of the central limit theorem with Johannes Hölzl and undergraduate student Luke Serafin in the Isabelle proof assistant. This involved, in particular, formalizing measure-theoretic probability and properties of characteristic functions [7]. A large scale project to formalize the Feit-Thompson theorem, led by George Gonthier, was completed in 2012 (Avigad contributed on a sabbatical year in France in 2009-2010), and the report was written and published during the grant period [4].

References

[1] Daniel R. Licata and Robert Harper, "Canonicity for 2-dimensional type theory," *ACM Principles of Programming Languages, 2012*.

[2] The Univalent Foundations Program, *Homotopy type theory: univalent foundations of mathematics,* open access and open source, http://homotopytypetheory.org/book/.

[3] Jeremy Avigad, "Uniform distribution and algorithmic randomness," *Journal of Symbolic Logi*, 78:334-344, 2013.

[4] Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, Fran cois Garillot, Stéphane Le Roux, Assia Mahboubi, Russell OConnor, Sidi Ould Biha, Ioana Pasca, Laurence Rideau, Alexey Solovyev, Enrico Tassi, Laurent Théry, "A machine-checked proof of the Odd Order Theorem," *Interactive Theorem Proving*, 2013.

DISTRIBUTION A: Distribution approved for public release

[5] Kuen-Bang Hou and Robert Harper. Covering Spaces. Talk presented at the TYPES Workshop at the Institut Henri Poincare in May of 2014.

[6] Carlo Angiuli, Ed Morehouse, Daniel R. Licata, and Robert Harper. *Homotopical Patch Theory. Internnational Conference on Functional Programming (ICFP) 2014*, Gothenburg, Sweden, June, 2014.

[7] Jeremy Avigad, Johannes Hölzl, and Luke Serafin, "A formally verified proof of the Central Limit Theorem (preliminary announcement)," Isabelle Workshop, Vienna, July 2014 (http://arxiv.org/abs/1405.7012).

[8] Jeremy Avigad, Robert Y. Lewis, and Cody Roux, "A heuristic prover for real inequalities," in Gerwin Klein and Ruben Gamboa, eds., *Interactive Theorem Proving 2014*, Springer, Heidelberg, 61-76, 2014.

[9] Jeremy Avigad and John Harrison, "Formally verified mathematics," *Communications of the ACM*, 57(4):66-75, 2014.

[10] Jeremy Avigad and José Iovino, "Ultraproducts and metastability," *New York Journal of Mathematics*, 19:713-727, 2013.

[11] Jeremy Avigad and Vasco Brattka, "Computability and analysis: the legacy of Alan Turing," in Rod Downey, editor, *Turing's Legacy: Developments from Turing's Ideas in Logic*, Cambridge University Press, 1-47, 2014.

[12] Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer, "The Lean theorem prover (system description)." *25th International Conference on Automated Deduction (CADE-25)*, Berlin, Germany, 2015.

[13] Leonardo de Moura, Jeremy Avigad, Soonho Kong, and Cody Roux, "Elaboration in dependent type theory, " `http://arxiv.org/abs/1505.04324`.

[14] Jeremy Avigad, Leonardo de Moura, Soonho Kong, Theorem Proving in Lean. `http://leanprover.github.io/tutorial/tutorial.pdf`.

[15] Jeremy Avigad, Krzystof Kapulkin, Peter LeFanu Lumsdaine, "Homotopy limits in type theory." *Mathematical Structures in Computer Science*, 25:1040-1070, 2015.

[16] Jeremy Avigad and Jason Rute, "Oscillation and the mean ergodic theorem for uniformly convex Banach spaces," *Ergodic Theory and Dynamical Systems*, 35(4):1009-1027, 2015.

[17] Jakob von Raumer, "Formalization of non-abelian topology for homotopy type theory

DISTRIBUTION A: Distribution approved for public release

[18] J. Tassarotti, D. Dreyer, and V. Vafeidis. “Verifying read-copy-update in a logic for weak memory.” In *Proc. of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) 2015*, pages 110-120. Repository at `http://www.cs.cmu.edu/~jtassaro/RCU/RCU.tar`.

[19] C. Angiuli, E. Morehouse, D. R. Licata, and R. Harper. “Homotopical Patch Theory (Revised and Expanded).” Under review, *Journal of Functional Programming*, Spring, 2015. Paper: `http://www.cs.cmu.edu/~cangiuli/papers/hpt-expanded.pdf`. Repository: `https://github.com/dlicata335/hott-agda/tree/homotopical-patch-theory-paper/programming`.

[20] G. Brunerie, E. Cavallo, K.-B. Hou (Favonia), N. Kraus, D. R. Licata and C Sattler. “HoTT Library in Agda.” `https://github.com/HoTT/HoTT-Agda.`

DISTRIBUTION A: Distribution approved for public release

1.

1. Report Type

Final Report

Primary Contact E-mail
Contact email if there is a problem with the report.

avigad@cmu.edu

Primary Contact Phone Number
Contact phone number if there is a problem with the report

412-268-8149

Organization / Institution name

Carnegie Mellon University

Grant/Contract Title
The full title of the funded effort.

Type Theory, Computation, and Interactive Theorem Proving

Grant/Contract Number
AFOSR assigned control number. It must begin with "FA9550" or "F49620" or "FA2386".

FA9550-12-1-0370

Principal Investigator Name
The full name of the principal investigator on the grant or contract.

Jeremy Avigad and Robert Harper

Program Manager
The AFOSR Program Manager currently assigned to the award

Tristan Nguyen

Reporting Period Start Date

07/15/2012

Reporting Period End Date

07/14/2015

Abstract

Type theory is a logical formalism that is rich enough to express complex mathematical and computational assertions. In this project, Avigad and Harper developed type-theoretic algorithms and formalisms that can support the development of secure and reusable software libraries, as well as the development of methods of automated reasoning in mathematics and libraries of mathematical knowledge.

The project is divided into two interrelated tracks. In the first track, Harper focused on extending contemporary type-theoretic frameworks to better support program development and verification. In the first track Harper supervised three separate mechanization projects, one using Coq, and two others using Agda. Student J. Tassarotti verified the Read-Copy-Update algorithm (used in the Linux kernel) using a logic for reasoning about weak memory models formulated in Coq. Student K.-B. Hou (Favonia) developed mechanizations of several results in homotopy theory (covering spaces, Seifert-van Kampen Theorem, Blakers-Massey Theorem) in Agda extended with higher inductive types and univalence. Collaborator D. R. Licata and student C. Angiuli mechanized in Agda the results described in their joint paper (with PI Harper and E Morehouse) "Homotopical Patch Theory".

In the second track, Avigad focused on interactive theorem proving and automated reasoning in specific

mathematical domains. With Krzysztof Kapulkin and Peter Lumsdaine, he developed a formal theory of homotopy limits, verified in the Coq interactive theorem prover. With PhD student Rob Lewis and postdoc Cody Roux, he has developed new heuristic, geometric methods of verifying real-valued inequalities. A python-based implementation has performed surprisingly well on benchmark problems from the verification of hybrid systems. Avigad is also centrally involved in work on a new interactive theorem prover, Lean, leading the development of Lean's formal libraries and supervising student work on the project. With Johannes Hölzl and Luke Serafin, he has also extended the library for measure-theoretic probability in the Isabelle theorem prover, and has verified a Fourier-analysis-based proof of the central limit theorem.

Distribution Statement

This is block 12 on the SF298 form.

Distribution A - Approved for Public Release

Explanation for Distribution Statement

If this is not approved for public release, please provide a short explanation. E.g., contains proprietary information.

SF298 Form

Please attach your SF298 form. A blank SF298 can be found here. Please do not password protect or secure the PDF The maximum file size for an SF298 is 50MB.

AFD-070820-035-Avigad-Harper.pdf

Upload the Report Document. File must be a PDF. Please do not password protect or secure the PDF . The maximum file size for the Report Document is 50MB.

final_report.pdf

Upload a Report Document, if any. The maximum file size for the Report Document is 50MB.

Archival Publications (published) during reporting period:

1) Daniel R. Licata and Robert Harper, "Canonicity for 2-dimensional type theory," ACM Principles of Programming Languages, 2012.
2) The Univalent Foundations Program, "Homotopy type theory: univalent foundations of mathematics," open access and open source, http://homotopytypetheory.org/book/.
3) Jeremy Avigad, "Uniform distribution and algorithmic randomness," Journal of Symbolic Logic, 78:334-344, 2013.
4) Georges Gonthier et al., "A machine-checked proof of the Odd Order Theorem," Interactive Theorem Proving, 2013.
5) Kuen-Bang Hou and Robert Harper. Covering Spaces. Talk presented at the TYPES Workshop at the Institut Henri Poincare in May of 2014.
6) Carlo Angiuli, Ed Morehouse, Daniel R. Licata, and Robert Harper. Homotopical Patch Theory. Internnational Conference on Functional Programming (ICFP) 2014, Gothenburg, Sweden, June, 2014.
7) Jeremy Avigad, Johannes Hölzl, and Luke Serafin, "A formally verified proof of the Central Limit Theorem (preliminary announcement), Isabelle Workshop, Vienna, July 2014 (http://arxiv.org/abs/1405.7012).
8) Jeremy Avigad, Robert Y. Lewis, and Cody Roux, "A heuristic prover for real inequalities," in Gerwin Klein and Ruben Gamboa, eds., Interactive Theorem Proving 2014, Springer, Heidelberg, 61-76, 2014.
9) Jeremy Avigad and John Harrison, "Formally verified mathematics," Communications of the ACM, 57(4):66-75, 2014.
10) Jeremy Avigad and José Iovino, "Ultraproducts and metastability," New York Journal of Mathematics, 19:713-727, 2013.
11) Jeremy Avigad and Vasco Brattka, "Computability and analysis: the legacy of Alan Turing," in Rod Downey, editor, Turing's Legacy: Developments from Turing's Ideas in Logic, Cambridge University Press, 1-47, 2014.
12) Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer, "The Lean theorem prover (system description)." 25th International Conference on Automated Deduction (CADE-25), Berlin, Germany, 2015.
13) Leonardo de Moura, Jeremy Avigad, Soonho Kong, and Cody Roux, "Elaboration in dependent type

theory," http://arxiv.org/abs/1505.04324.
14) Jeremy Avigad, Leonardo de Moura, Soonho Kong, Theorem Proving in Lean. http://leanprover.github.io/tutorial/tutorial.pdf.
15) Jeremy Avigad, Krzystof Kapulkin, Peter LeFanu Lumsdaine, "Homotopy limits in type theory."Mathematical Structures in Computer Science, 25:1040-1070, 2015.
16) Jeremy Avigad and Jason Rute, "Oscillation and the mean ergodic theorem for uniformly convex Banach spaces," Ergodic Theory and Dynamical Systems, 35(4):1009-1027, 2015.
17) J. Tassarotti, D. Dreyer, and V. Vafeidis. Verifying read-copy-update in a logic for weak memory. In Proc. of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI) 2015, pages 110-120. Repository at url http://www.cs.cmu.edu/~jtassaro/RCU/RCU.tar.
18) C. Angiuli, E. Morehouse, D. R. Licata, and R. Harper. Homotopical Patch Theory (Revised and Expanded). Under review, Journal of Functional Programming, Spring, 2015. Paper: http://www.cs.cmu.edu/~cangiuli/papers/hpt-expanded.pdf. Repository: https://github.com/dlicata335/hott-agda/tree/homotopical-patch-theory-paper/programming.
19) G. Brunerie, E. Cavallo, K.-B. Hou (Favonia), N. Kraus, D. R. Licata and C Sattler. HoTT Library in Agda. URL: https://github.com/HoTT/HoTT-Agda.

Changes in research objectives (if any):

(none)

Change in AFOSR Program Manager, if any:

(none)

Extensions granted or milestones slipped, if any:

(none)

AFOSR LRIR Number

LRIR Title

Reporting Period

Laboratory Task Manager

Program Officer

Research Objectives

Technical Summary

Funding Summary by Cost Category (by FY, $K)

	Starting FY	FY+1	FY+2
Salary			
Equipment/Facilities			
Supplies			
Total			

Report Document

Report Document - Text Analysis

Report Document - Text Analysis

Appendix Documents

2. Thank You

E-mail user

Aug 29, 2015 17:50:37 Success: Email Sent to: avigad@cmu.edu